

Debrief

Cruise is a Google Chrome Extension that utilizes user keyboard text input and pupil dilation/EEG brain data to increase writing productivity. It predicts when the user will approach a writing roadblock. A roadblock is defined as the user not maintaining their writing productivity thresholds. Custom writing productivity thresholds can be specified by the user; these thresholds are word count and page count. When a roadblock occurs the app offers user-specific solutions such as recommending research articles, scaffolding questions (both of which are dependent on the user's written content), or taking a break. Additionally, the product takes note of the quality and quantity of writing through tracking the user's output (i.e. word count, page count, etc.) as well as their creativity through the Linguistic Inquiry and Word Count (LIWC) and Creativity Corpus Keywords Dictionary (i.e. cognitive, emotional, and structural elements as well as level of creativity and whether the writer is displaying creativity in the scientist or artist domain). Furthermore, a user-specific break model will be developed through keeping track of when, how many, and at what frequency breaks are taken. The user writes in a Google Doc while the Chrome Extension processes their data, providing updates and guidance in real time.

Done

Set for later

In progress

Not started

September 30 11:59 PM: deadline for application to Incubator Program

October 18: Incubator Program begins

Timeline

- Ensure Princeton accounts work - permissions issue because administrator also has access
- Fix so don't need to transfer document link to new tab
- Measure page breaks using Google Docs API

May 30-June 12:

- Connect Javascript document URL variable to Python keyboard features file
- Pass variable in keyboard features file to automate retrieving the document ID

June 13-June 30:

- Send Javascript total time variable to Python for machine learning logic
- Develop notifications for roadblock
- Develop keyboard machine learning logic

June 30 - July 10:

- Develop notifications for completion
- Develop completion logic
- Develop estimated completion time logic
- Define labels: roadblock and completion time
- Collect keyboard data
- Train keyboard machine learning algorithm

July 10 - July 27:

- Finish completion ml logic and model
- Choose between pupil and brain data
- Develop continuously updated ml based on user data

July 28 - August 10:

- Fix roadblock/standby predictions

- Ensure ML model is accurate

August 10 - August 31:

- Organize multiprocessing of all components
- Get URL without selenium
- Connect machine learning to chrome extension
- Roadblock and completion notifications with service workers
- Deploy product

September 5:

- Send MVP to Princeton student community through Hoagie and ask each participant to fill out mandatory survey after using product
- Keep track of responses and close in either mid or end September
- End September:
 - Analyze pilot study data
 - Depending on outcome determine what is free and what is part of subscription as well as how much subscription is
 - Reapply PSV showing successful pilot study for funding

September 6 - end of semester:

- Develop brain data engineering
- Develop brain machine learning logic
- Collect brain data
- Train brain machine learning algorithm
- Pair keyboard and brain machine learning algorithms
- Add paired functionalities to Chrome Extension
- Research creativity models (LIWC and Creativity Corpus Keywords Dictionary)
- Determine pre existing software we can use and payment
- Develop creativity data engineering
- Develop creativity machine learning algorithm
- Add creativity functionalities to Chrome Extension

Throughout academic school year:

- Complete Incubator Program
- Apply for more funding
- Continue applying to entrepreneurship competitions
- Develop new idea or continuation of present idea
- Send to other universities, schools, and newspaper and magazine companies

User Inputs: total time user plans to spend on assignment, wordcount, pagecount, font settings (font family, font size)

Features (ML inputs): charcount, wordcount, sentence count, standby, standby number, number of roadblocks, variations of these (produced, deleted, change in)

How roadblock is calculated: charcount, wordcount, sentence count rate less or same as value calculated in previous 5 minutes, standby

Labels (ML Outputs): roadblock logic, how long user would usually spend on assignment based on previous data

User Outputs: prediction of how long user will take to complete assignment (before they start, based on previous work session), history log of when roadblocks were displayed, final completion notification

Data from previous session: how long they spent, all data in keyboard_training_features, history log of roadblocks

Ideas: remaining time and effort needed if below threshold (differences), google calendar functionality